MIDTERM REVIEW

& quiz section assignment

Short answer: scoring

- For multipart questions:
  - 1 answer: 1 point
  - 2 answers: 3 points
  - 3 answers: 6 points
- For final question:
  - 2 points for definition
  - 1, 1, 2 points for examples

Short answer: questions

- Define coupling
  - Dependency between modules, processes, things
- Give 3 examples of loose coupling including 1 from your project.
  - MVC, the web, Unix pipes/filter, donation collection/distribution
- What is a *metric* for a project (as we’ve defined in class)?
  - Quantifiable measure of success

Short answer: questions

- What is the principle difference between a staged delivery lifecycle and evolutionary delivery lifecycle?
  - Staged: requirements/arch rigidly defined
  - Evolutionary: iterates on these
- Give 2 examples of why using UML is good. Give 1 example of why using UML is bad.
  - +: clear, language agnostic, makes your boss happy
  - -: time consuming, incomplete
- Data crunching application. What are 3 general criteria you would use to make this decision?
  - Does the job, robust community, maturity, familiar
  - -ilities or modularity concepts

Short answer: questions

- 3 disadvantages to using a Waterfall Lifecycle process?
  - Need up-front knowledge, can’t swim upstream, slow to deliver, rigid, linear, hard to pipeline
- Give 3 reasons why adding people to a team might make the team more productive.
  - Division of labor, specialization, redundancy, water-cooler effect
- Define “divide and conquer” as it relates to a strategy. Give 3 specific examples, including 1 from your project.
  - Breaking down a large problem into smaller problems and solving those
  - Ex: team organization, front end vs backend, MVC, house project, layer cake architecture

Assignment: URL Validation

http://en.wikipedia.org/wiki/URI_scheme#Generic_syntax

- What is a “valid” url? There are regex for this.
- Any URL or just http?
  - mailto:me@example.com?
- What about:
  - Fragments?
  - Port name?
  - Query string?

URL Comparison

- How do we compare URLs to sort?
- Need to standardize them first: http://en.wikipedia.org/wiki/URL_normalization
- Decide how to determine >, <, ==
- Example: https://mail.google.com/mail/u/0/#label/CSE403
    - Consider http vs https?
    - Consider www?
    - Would this be equivalent to https://mail.google.com/mail/? >?